



توسعه برنامه‌های اندرویدی دیگر همچون گذشته یک رویکرد ساده مبتنی بر یک زبان محبوب و چند محیط توسعه یکپارچه نیست. امروزه، به روش‌های مختلفی می‌توان برنامه‌های اندرویدی را ایجاد کرد؛ حتی بدون نیاز به دانش برنامه‌نویسی و از طریق سرویس‌های آنلاین، با محدودیت‌هایی در این زمینه روبه‌رو هستید. اما اگر به دنبال ساخت برنامه‌های جدی هستید، هیچ جایگزینی بهتر از زبان‌های برنامه‌نویسی نیست. اگر به دنبال آن هستید تا آینده خوبی را در اکوسیستم اندروید برای خود رقم بزنید باید به دنبال یادگیری یک زبان روبه‌رشد و جدید باشید. زبانی که انعطاف‌پذیری بالایی داشته و بهترین قابلیت‌های زبان‌های مختلف را در خود جای داده باشد. کوتلین، یکی از بهترین گزینه‌های پیش رو است. زبانی که بر آینده توسعه برنامه‌های اندرویدی و اکوسیستم اندرویدی تأثیرگذار است.

اگر بگوییم امروز یک‌زمان رویایی برای توسعه‌دهندگان اندرویدی است، اغراق نکرده‌ایم. دلیل این ادعا به پشتیبانی رسمی گوگل از کوتلین بازمی‌گردد. کوتلین، ویژگی‌هایی را که توسعه‌دهندگان برای سال‌های متمادی به دنبال آن‌ها بودند، در اختیارشان قرار داده است. یک زبان [برنامه‌نویسی Statically-Typed](#) که می‌تواند روی ماشین مجازی جاوا اجرا شود. زبانی متن‌باز که اجازه می‌دهد از دو پارادایم برنامه‌نویسی شیء‌گرا و تابعی برای ساخت برنامه‌ها استفاده کنید. این زبان نخستین بار در سال 2011 میلادی از سوی شرکت روسی JetBrains به دنیای نرم‌افزار معرفی شد. زبانی که از سوی توسعه‌دهندگان مختلفی بسط داده شد و قابلیت‌های کاربردی به آن اضافه شد.

گوگل دیگر جاوا را دوست ندارد

امروزه، کوتلین به دو دلیل موردتوجه است: اول آن‌که راه‌حل‌های ارائه‌شده از سوی این زبان پاسخی است به مشکلات چندساله توسعه‌دهندگان اندرویدی؛ دوم آن‌که گوگل به‌عنوان یکی از بزرگ‌ترین شرکت‌های نرم‌افزاری از آن پشتیبانی می‌کند. ایمنی، خوانایی، ارائه راه‌حل‌های کاربردی، حل مشکلات رایج جاوا و قابلیت همکاری متقابل با زبان‌های دیگر هر توسعه‌دهنده‌ای را متقاعد می‌سازد که به سمت گزینه بهتر برود. درحالی‌که گوگل در کنفرانس توسعه‌دهندگان I/O سال گذشته حمایت رسمی خود را از این زبان اعلام کرد، اما اکنون این شرکت به دنبال آن است که پلتفرم اندروید و طراحی برنامه‌ها را برای این پلتفرم به سمت کوتلین سوق دهد. گوگل، به‌طور ضمنی به توسعه‌دهندگان بزرگ اعلام داشته، کوتلین را جایگزین جاوا کنند، به همین دلیل است که امروزه شاهد هستیم طیف گسترده‌ای از شرکت‌ها و توسعه‌دهندگان در حال بازنویسی برنامه‌های جاوا خود به زبان کوتلین هستند. رویکردی که در درازمدت کوتلین را به زبان اصلی ساخت برنامه‌های اندرویدی تبدیل خواهد کرد. مشابه چنین رویکردی را پیش‌تر از سوی اپل و در ارتباط با زبان برنامه‌نویسی سوئیفت و زبان آبجکت‌یوسی شاهد بودیم. تبلیغات مؤثر از یک سو و کارآمدی کوتلین، در حال به حاشیه راندن جاوا در اکوسیستم توسعه برنامه‌های اندرویدی است. موجی که از یک سال پیش بلند شده اکنون به‌تدریج در حال بلندتر شدن است تا کوتلین را به بازیگر اصلی دنیای اندروید تبدیل کند. درست از همان زمانی‌که گوگل اعلام کرد کوتلین را برای ساخت برنامه‌های اندرویدی به رسمیت شناخته و

خود نیز در مسیر توسعه **کوتلین** گام برخواهد داشت، این زبان را در کانون توجه توسعه‌دهندگان قرار داد. دلیل اصلی برای این اقدام گوگل مدرن بودن **کوتلین** است. قدرت بالا در کدنویسی مختصر و شفاف، انعطاف‌پذیری، ارائه گلچینی از بهترین قابلیت‌های موجود در سایر زبان‌های برنامه‌نویسی و تعامل مستقیم با جاوا باعث شده که شما به‌عنوان یک توسعه‌دهنده در مدت‌زمان کوتاهی بتوانید برنامه کاربردی خود را با این زبان بسازید. نکته جالب توجه آن‌که کارشناسان حوزه نرم‌افزار اعلام داشته‌اند، حمایت اوراکل از جاوا و توسعه زبان جاوا هیچ‌تأثیر منفی روی **کوتلین** نخواهد گذاشت، به دلیل این‌که **کوتلین** زبانی است که بر پایه ایده‌های به دست آمده از جامعه برنامه‌نویسان ساخته شده است. برنامه‌نویسانی که هیچ‌گاه موفق نشدند برای مشکلاتی که در جاوا داشتند، پاسخ و راه‌حل درستی به دست آورند.



چالش‌های کدنویسی در کوتلین برطرف شده است

کوتلین بیشتر مشکلاتی را که توسعه‌دهندگان در زبان جاوا با آن دست‌به‌گریبان بودند، برطرف کرده است. هر چند **کوتلین** هم مشکلات خاص خود را دارد. یکی از نخستین چالش‌هایی که برنامه‌نویسان **کوتلین** با آن روبه‌رو بودند در ارتباط با نحوه تعریف و کار با مقادیر تهی بود. سرانجام پس از ارائه راه‌حل‌های متعدد این تیم موفق شد بر این مشکل غلبه کند.

در زبان **کوتلین** شما دو نوع تهی‌پذیر (nullable) و غیر تهی (Non-Null) در اختیار دارید. اگر متغیری، غیرتهی‌پذیر تعریف شود، در عمل نمی‌توانید مقدار تهی به آن اختصاص دهید. در چنین شرایطی کامپایلر به شما این مسئله را گوشزد می‌کند تا مانع بروز خطاهای زمان اجرا شود. همچنین برای تعریف مقدار تهی‌پذیر در **کوتلین** باید از سمبل (?) در زمان تعریف متغیر استفاده کنید.

مطلب پیشنهادی



تجربه‌ای جدید در برنامه‌نویسی
راهنمای شروع به کار با زبان برنامه‌نویسی Kotlin

مشکلات جاوا در کوتلین برطرف شده‌اند

کوتلین بیشتر مشکلات موجود در زبان جاوا را برطرف کرده است. برای مثال، ارجاع تهی در **کوتلین** از طریق سیستم نوع‌ها مدیریت می‌شود. **کوتلین** از راهکارها و توابع قدرتمندی برای این منظور استفاده می‌کند. آرایه‌ها نیز در **کوتلین** غیرمستقیم هستند. سیستم مدیریت و نظارت بر استثناءها در **کوتلین** متفاوت از جاوا است.

مزایای کلیدی کوتلین

سازمان‌ها به راحتی **کوتلین** را برای ساخت برنامه‌های کاربردی اندروید به رسمیت شناخته و از آن استفاده می‌کنند. اما توسعه‌دهندگان سنتی که طی این سال‌ها از جاوا استفاده کرده‌اند، به دلایل متقاعدکننده‌ای نیاز دارند تا **کوتلین** را جایگزین جاوا کنند. در ادامه به ذکر مهم‌ترین دلایلی می‌پردازیم که شما را متقاعد می‌سازند **کوتلین** را جایگزین

جاوا کنید.

1. کوتلین مختصرنویسی را پیشنهاد می‌کند

مختصرنویسی و کوتاه بودن مورد قبول همگان است. مختصر بودن کدهای **کوتلین** یکی از اصلی‌ترین دلایلی است که توسعه‌دهندگان جاوا را به سمت خود جذب می‌کند. جاوا در این زمینه رویکردی متضاد دارد. زبانی که بیش از اندازه برای انجام کارها به کدنویسی طولانی علاقه دارد. خلاصه‌نویسی تضمین می‌کند در کدهای نوشته‌شده خطای کمتری وجود خواهد داشت. مختصرنویسی به میزان قابل‌توجهی بهره‌وری را افزایش می‌دهد.

2. متن‌باز است

شرکت JetBrains از سال 2012 به بعد تصمیم گرفت پروژه **کوتلین** را متن‌باز کند. یک زبان برنامه‌نویسی متن‌باز دو ویژگی مثبت برای یک توسعه‌دهنده به همراه می‌آورد: اول اجازه می‌دهد کدهایی باکیفیت بالا بنویسد؛ دوم اطمینان می‌دهد که تیم پشتیبانی‌کننده **کوتلین** به بهترین شکل از او حمایت خواهند کرد. تیمی که حتی ممکن است متشکل از توسعه‌دهندگان سراسر جهان باشد.

3. یک زبان بالغ بوده و پشتیبانی یکی از بهترین محیط‌های توسعه برنامه‌نویسی را به همراه دارد

کوتلین پیش از آن‌که به شکل یک زبان حرفه‌ای به بازار عرضه شود برای مدت‌های طولانی در وضعیت آلفا و بتا قرار داشت. به این معنا که پیش از انتشار رسمی، در قالب چند پروژه در اختیار توسعه‌دهندگان قرار گرفت. این کار به دو دلیل انجام شد: اول آن‌که بهترین ویژگی‌های زبان‌های برنامه‌نویسی که عملکرد خوب آن‌ها به اثبات رسیده، به این زبان وارد شود؛ دوم آن‌که پشتیبانی بهترین محیط توسعه یکپارچه را به دست آورد. رویکردی که تا حد زیادی در زبان جولیا نیز شاهد آن بوده‌ایم. فراموش نکنید که شما می‌توانید از کوتلین در محیط توسعه یکپارچه **اندروید** استودیو استفاده کنید. محیطی که شبیه به ویژوال استودیو مایکروسافت بهترین ویژگی‌ها و ابزارها را چه در زمینه کدنویسی و چه در زمینه خطایابی و مدیریت کدها در اختیارتان قرار می‌دهد.

4. ساده‌ترین راهکار را برای توسعه برنامه‌های اندرویدی ارائه می‌کند

کوتلین زبان قدرتمندی است که یادگیری ترکیب نحوی آن در دسرهای زبان جاوا را ندارد. اگر جاوا در ارتباط با یکسری مسائل سخت‌گیر بوده، اسکالا در ارتباط با کامپایل برنامه‌ها کند عمل می‌کند، ترکیب نحوی C و ++C پیچیده است، کوتلین این چنین مشکلاتی را ندارد. در **کوتلین** برای نوشتن برنامه‌های **اندرویدی** به کتابخانه‌های کوچکی نیاز دارید که این کتابخانه‌ها نه تنها روند ساخت برنامه‌ها را ساده می‌کنند، بلکه سرعت انجام کارها را نیز افزایش می‌دهند.

5. کوتلین، سوئیفت توسعه برنامه‌های اندرویدی است

کوتلین به‌طور کامل با **اندروید** استودیو یکپارچه‌شده و همین مسئله سرعت آن را افزایش داده است. یک پروژه **کوتلین** در کمتر از 10 دقیقه می‌تواند تنظیم‌شده و عملیاتی شود. **اندروید** 2.3 تنها به یک پلاگین نیاز دارد تا به شکل یکپارچه **کوتلین** را در اختیار شما قرار دهد، درحالی‌که اندروید 3.0 به‌طور کامل با **کوتلین** یکپارچه شده است. مانند جاوا، **کوتلین** از طریق محیط توسعه یکپارچه قابل‌استفاده است. کدهای این زبان را بدون هیچ‌گونه مشکلی می‌توانید خطایابی کنید. در زبان **کوتلین** به همان ویژگی‌هایی دسترسی دارید که در جاوا برای انجام بسیاری از کارها از آن‌ها استفاده می‌کنید. در این زمینه **کوتلین** عملکرد خوبی داشته است.

6. روند روبه رشد و تکاملی کوتلین نشان می‌دهد که زبان قابل‌اطمینانی است

یکی از دلایل اصلی شکل‌گیری زبان **کوتلین** به رویکرد شرکت JetBrains بازمی‌گردد. این شرکت برای انجام کارهای خود یکسری قابلیت‌های کلیدی در اختیار داشت. ویژگی‌های کاربردی که از آن‌ها برای پیشبرد پروژه‌های خود استفاده می‌کرد. همین مسئله توسعه‌دهندگان این شرکت را به فکر انداخت تا به توسعه ویژگی‌هایی بپردازند که در اختیار داشتند. با پشتیبانی گوگل، هر دو شرکت روی ایده‌های نوآورانه متمرکز شدند. **کوتلین** نه فقط به دلیل تلاش‌های قابل‌تحسینی که در زمینه مفهوم‌سازی اولیه در تاریخ خود ارائه کرد، شهرت پیدا کرد، بلکه به دلیل ارائه یکسری ایده‌های نوینی که انتظار می‌رود در آینده موفقیت‌های قابل‌توجهی را عاید این زبان کند، مورد ستایش قرار

گرفته است. دور از انتظار نیست اگر فکر کنیم گوگل از آن جهت از **کوتلین** حمایت کرد که اطمینان داشت آینده درخشانی را هم برای خود (**کوتلین**) و هم برای **اندروید** رقم خواهد زد. قابلیت‌هایی همچون Kotlin Android Extensions، کتابخانه‌هایی همچون Anko، بهینه‌سازی مستمر، حجم کم کتابخانه‌ها و کم کردن زمان ساخت نشان از آن دارند که JetBrains خود را متعهد به توسعه **کوتلین** می‌داند. رویکردی که در عمل به نفع **اندروید** تمام خواهد شد.

7. بینش شما را عمیق می‌کند

زمانی که توسعه‌دهندگان تصمیم گرفتند از **کوتلین** استفاده کنند، مشاهده کردند که به‌کارگیری یک زبان مدرن تا چه اندازه فرآیندهای کاری را ساده‌تر کرده و باعث شکوفایی ایده‌های بکر می‌شود. نکته جالب توجه آن‌که برخی از توسعه‌دهندگان بر این باورند، دیگر عمر زبان جاوا به انتها رسیده که البته این حرف چندان درست نیست. جاوا هنوز هم در بسیاری از حوزه‌ها پر کاربرد بوده و قدرتمند است. اما به‌کارگیری یک زبان جدید و مدرن به شما اجازه می‌دهد متفاوت ببانید. متفاوت اندیشیدن از آن جهت برای یک توسعه‌دهنده مهم است که انعکاس مستقیمی روی کار او خواهد گذاشت. از دیدگاه روان‌شناسی اشتیاق به فهمیدن موضوعات و تفکر در ارتباط با آن‌ها شما را ترغیب می‌کند، به سراغ یادگیری مباحث (زبان‌های) مشابه بروید. در مجموع این شما هستید که در پروفایل و رزومه کاری خود گنجینه‌ای از مهارت‌ها را درج خواهید کرد.

8. ایمن‌تر از جاوا است

کوتلین به لحاظ امنیت بهتر از جاوا است. یکی از مشکلات عمده زبان جاوا به طراحی خود این زبان بازمی‌گردد که در عمل توسعه‌دهنده را مجبور می‌کند برای ساخت یک برنامه کار اضافی انجام دهد. همین مسئله باعث می‌شود تا پیاده‌سازی پروژه‌ها زمان‌بر شده و مشتری مجبور شود پول بیشتری را پرداخت کند. مقادیر تهی عامل بروز بسیاری از خطاهای نرم‌افزاری در جاوا هستند. پژوهشی که چندی پیش در همین ارتباط انجام شد، نشان داد در بیشتر از 90 درصد موارد خطاهای رخ داده در برنامه‌های جاوا از نوع Null Pointer Exception هستند. درحالی‌که در **کوتلین** شما با چنین مشکلی روبه‌رو نیستید. در **کوتلین** توسعه‌دهنده مجبور است متغیرهایی که مقدار تهی را قبول می‌کنند با علامت (?) مشخص کند.

مطلب پیشنهادی



کوتاه بنویسید تا سریع‌تر اجرا شوند
این 18 تکنیک کوتاه‌سازی کدهای جاوا اسکرپت، شما را شگفت‌زده می‌کند

چرا جاذبه کوتلین کسب‌وکارها را به سمت خود می‌کشد؟

برای یک کسب‌وکار حرکت به سمت یک ابزار جدید تصمیمی خردمندانه است. هر ابزار جدیدی با هدف ارائه راهکاری برای حل یک مشکل به شیوه‌ای ساده و بهتر عرضه می‌شود. یکی از دلایلی که باعث شده کسب‌وکارها به **کوتلین** به‌منظور ساخت برنامه‌های **اندرویدی** روی خوش‌نشان دهند، به حمایت رسمی گوگل از این زبان بازمی‌گردد. این پشتیبانی به معنای آن است که فرایند توسعه برنامه‌های **اندرویدی** در آینده‌ای که تجهیزات مبتنی بر **اندروید** عملکرد پیچیده‌تری به خود خواهند گرفت، ساده شود. سهولت یادگیری زبان به معنای آن است که کارمندان یک شرکت برای یادگیری **کوتلین** نیازی ندارند وقت زیادی را صرف کنند. توسعه و نگهداری پروژه‌ها با هزینه کمتر یک مزیت بزرگ برای کسب‌وکارها به همراه می‌آورد.

هر چه تعداد کسب‌وکارهایی که از **کوتلین** استفاده می‌کنند، بیشتر شود، گوگل و انجمن‌های مرتبط با **اندروید** پشتیبانی بیشتری از **کوتلین** به عمل خواهند آورد و تاکید کمتری بر جاوا خواهند داشت. با توجه به این‌که جاوا در حال حرکت به سمت الگوی پرداختی است، **کوتلین** به‌عنوان یک پروژه متن‌باز جایگزین خوبی برای آن خواهد بود.

تاریخ انتشار:

نشانی منبع:

<https://www.shabakeh-mag.com/application/14301/%DA%A9%D9%88%D8%AA%D9%84%DB%8C%D9%80%D9%86-%D9%BE%D8%B1-%D9%BE%D8%B1%D9%88%D8%A7%D8%B2-%D8%A7%D9%86%D8%AF%D8%B1%D9%88%DB%8C%D8%AF>